REMARKS

This Preliminary Amendment is presented to clearly and distinctly claim the invention. No new matter is added. Entry is respectfully requested.

By this amendment, Claims 1 - 44 have been cancelled. Claims 45-96 have been amended. Hence, Claims 45-96 are pending in the application.

It is respectfully submitted that all of the pending claims are in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any fee shortages or credit any overages Deposit Account No. 50-1302.

Respectfully submitted,
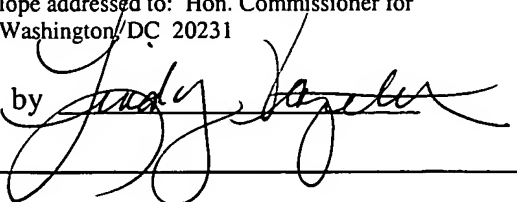
HICKMAN PALERMO TRUONG & BECKER LLP

Christopher J. Brokaw
Reg. No. 45,620

1600 Willow Street
San Jose, CA 95125
(408) 414-1080, ext. 207
Date: November 8, **2002**
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Hon. Commissioner for Patents, BOX NON-FEE AMEND, Washington, DC 20231

on Nov 8, 2002   by _____

11

Application of Carol L. Colrain, et al., Ser. No. 10/086,842, Filed February 28, 2002
Separate Paper with Version with Markings to Show Changes Made to and Claims to Accompany
Preliminary Amendment

VERSION WITH MARKINGS TO SHOW CHANGES MADE TO SPECIFICATION

For purposes of showing the changes from the current version of the application, deleted text is shown in ~~strikethrough~~ and added text is shown in **bold**.

Paragraph on page 6, lines 6-15

FIGURE 1 is a block diagram showing a cluster database 12 incorporating high availability components. Parallel database servers 11a-d, each including a cooperative resource group are each coupled to a single cluster database 12 to form a high availability cluster framework 14, such as described in **pending** U.S. Patent Application, Serial No. **10/086,782** ———, entitled "System and Method For Providing Cooperative Resource Groups For High Availability Applications," ~~Attorney Docket No. 007.0191.01,~~ filed February, 28, 2002, ~~pending,~~ the disclosure of which is incorporated by reference. The servers 11 process a stream of transactions received from clients, such as client 13 and remote client 18, in parallel with each server processing an entire transaction.

Paragraph on page 6, lines 23-29

The cluster framework 14 appears as a single node to individual clients, which subscribe to the servers published by each cluster. The client sessions receive notification of any changes in the services provided, such as described in **pending** U.S. Patent Application, Serial No. **10/086,949** ———, entitled "system and Method For Providing Out-Of-Band Notification of Service Changes," ~~Attorney Docket No. 007.0192.01,~~ filed February 29, 2002, ~~pending,~~ the disclosure of which in incorporated by reference, and transfer to alternate nodes as necessary.

Application of Carol L. Colrain, et al., Ser. No. 10/086,842, Filed February 28, 2002
Separate Paper with Version with Markings to Show Changes Made to and Claims to Accompany
Preliminary Amendment

Paragraph starting on page 6, line 30 – page 7, line 11

Within each cluster framework 14, each of the database servers 11 incorporate
high availability components, such as described in J. Gray et al., "Transaction Processing:
Concepts and Techniques," pp. 128-38, M. Kaufmann Pubs., San Francisco, California
(1993), the disclosure of which is incorporated by reference. Failover processing is
initiated upon the detection of the termination of **a** ~~an~~ database instance, such as
described in **pending** U.S. Patent Application, Serial No. **10/087,494** ——, entitled
"System and Method For Detecting Termination Of An Application Instance Using
Locks," ~~Attorney Docket No. 007.0194.01,~~ filed February 28, 2002, ~~pending,~~ the
disclosure of which is incorporated by reference. Likewise, upon a planned shutdown, an
application will switch over to another instance of the database supporting the service.
Other situations in which failover processing is required are possible, as would be
recognized by one skilled in the art.

Paragraph starting on page 7, line 26 – page 8, line 8

FIGURE 2 is a functional block diagram showing a database stack 31 implemented on a server node 30, including a system for detecting termination of **a** ~~an~~ database instance using locks, such as described in **pending** U.S. Patent Application, Serial No. **10/087,494** ———, entitled "System and Method For Detecting Termination Of An Application Instance Using Locks," ~~Attorney Docket No. 007.0191.04,~~ filed February 28, 2002, ~~pending,~~ the disclosure of which is incorporated by reference. The database stack 31 is logically divided into two parts: a cooperative resource group 32, and a resource 33. The cooperative resource group 32 includes a mobile internet protocol (IP) address 36, a database instance 35 (or high availability application), and external monitors 34. The mobile IP address 36 is assigned to the cooperative resource group 32 to support client access. More generally, a generic high availability application could execute within the cooperative resource group 32, instead of the database instance 35, as would be recognized by one skilled in the art.

VERSION WITH MARKINGS TO SHOW CHANGES MADE TO CLAIMS

All pending claims are reproduced below in marked-up form, whether or not amended, for the convenience of examination.

45.   (New) A method for processing a request, comprising the steps of:

storing a first cursor in a first node, said first cursor comprising a shareable part and a non-sharable part;

establishing a second cursor based on the shareable part of the first cursor;

retaining the second cursor at the second node, wherein the second cursor is not executed while said first node continues to be able to execute said first

cursor; and

causing said client to interact with said second node in response to said first node ceasing to be able to execute said first cursor.

46. (New) The method of claim 45, wherein the first node is a first database server, and the second node is a second database server.

47. (New) The method of claim 45, further comprising the step of executing said second cursor at said second node after the first node ceases to be able to execute said first cursor.

48. (New) The method of claim 47, wherein a non-sharable part of said second cursor is created upon execution of said second cursor.

49. (New) The method of claim 45, further comprising the step of transmitting the shareable part of the first cursor to the second node in anticipation of said first node not being able to execute said first cursor.

50. (New) The method of claim 45, further comprising the step of transmitting the shareable part of the first cursor to the second node prior to an indication that said first node will be unable to execute said first cursor.

51. (New) The method of claim 45, further comprising the step of transmitting the shareable part of the first cursor to the second node in response to a shutdown of said first node.

52. (New) The method of claim 45, further comprising the step of recreating at said second node at least one of recursive cursors and system cursors.

15

53.   (New) The method of claim 45, further comprising the step of extracting said sharable portion of said first cursor.

54.   (New) The method of claim 53, wherein the step of extracting said sharable portion of said first cursor extracts data of at least one type, wherein said type is a member of a group consisting of bind variables data, session environment data, language version data, object authorization data, object names data, SQL text string data, and description data.

55.   (New) The method of claim 45, wherein the step of compiling the shareable part of the first cursor to form a second cursor at the second node comprises: generating an execution plan for an operation associated with said second cursor.

56.   (New) The method of claim 45, wherein the step of compiling the shareable part of the first cursor to form a second cursor at the second node comprises: generating a parse tree for an operation associated with said second cursor.

57.   (New) A method for processing a request, comprising the steps of:
initiating execution of a database statement at said first database server;
transmitting said database statement to a second database server prior to
        termination of execution of said database statement at said first database
        server; and
generating, at said second database server, a structure required to prepare said
        database statement for execution at said second database server.

58.   (New) The method of claim 57, further comprising the step of receiving a request from a client at said second database server to execute said database statement.

59.   (New) The method of claim 58, further comprising the step of executing said database statement at said second database server using said structure.

60.   (New) The method of claim 57, wherein said structure is a parse tree.

61.   (New) The method of claim 57, wherein said structure is an execution plan.

62.   (New) The method of claim 57, wherein said step of transmitting said database statement to a second database server is performed in anticipation of a shutdown of said first database server.

63.   (New) The method of claim 57, wherein said step of transmitting said database statement to a second database server is performed prior to any indication that said first database server may become inoperable.

64.   (New) The method of claim 57, wherein said step of transmitting said database statement to a second database server comprises the step of:
determining that said second database server is capable of executing said database statement.

65.   A method processing a request, comprising the steps of:
receiving a database statement to be executed at a first database server;
generating a structure required to prepare the statement for execution at said first database server;
initiating execution of said database statement at said first database server;
transmitting said structure to a second database server;
receiving a request to execute said database statement at said second database server; and
executing said database statement at said second database server using said structure.

66.   (New) The method of claim 65, wherein said structure is a parse tree.

67.   (New) The method of claim 65, wherein said structure is an execution plan.

17

68.   (New) The method of claim 65, wherein said step of transmitting said structure to a second database server is performed in anticipation of a shutdown of said first database server.

69.   (New) The method of claim 65, wherein said step of transmitting said structure to a second database server is performed prior to any indication that said first database server may become inoperable.

70. (New) The method of claim 65, wherein said step of transmitting said structure to a second database server comprises the step of:
determining that said second database server is capable of executing said database statement.

71 (New) A computer-readable medium carrying one or more sequences of instructions for processing a request, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

storing a first cursor in a first node, said first cursor comprising a shareable part and a non-sharable part;

establishing a second cursor based on the shareable part of the first cursor;

retaining the second cursor at the second node, wherein the second cursor is not executed while said first node continues to be able to execute said first cursor; and

causing said client to interact with said second node in response to said first node ceasing to be able to execute said first cursor.

72. (New) The computer-readable medium of claim 71, wherein the first node is a first database server, and the second node is a second database server.

73. (New) The computer-readable medium of claim 71, further comprising the step of executing said second cursor at said second node after the first node ceases to be able to execute said first cursor.

74. (New) The computer-readable medium of claim 73, wherein a non-sharable part of said second cursor is created upon execution of said second cursor.

19

75. (New) The computer-readable medium of claim 71, further comprising the step of transmitting the shareable part of the first cursor to the second node in anticipation of said first node not being able to execute said first cursor.

76. (New) The computer-readable medium of claim 71, further comprising the step of transmitting the shareable part of the first cursor to the second node prior to an indication that said first node will be unable to execute said first cursor.

77. (New) The computer-readable medium of claim 71, further comprising the step of transmitting the shareable part of the first cursor to the second node in response to a shutdown of said first node.

78. (New) The computer-readable medium of claim 71, further comprising the step of recreating at said second node at least one of recursive cursors and system cursors.

79. (New) The computer-readable medium of claim 71, further comprising the step of extracting said sharable portion of said first cursor.

80. (New) The computer-readable medium of claim 79, wherein the step of extracting said sharable portion of said first cursor extracts data of at least one type, wherein said type is a member of a group consisting of bind variables data, session environment data, language version data, object authorization data, object names data, SQL text string data, and description data.

81. (New) The computer-readable medium of claim 71, wherein the step of compiling the shareable part of the first cursor to form a second cursor at the second node comprises:
generating an execution plan for an operation associated with said second cursor.

82.   (New) The computer-readable medium of claim 71, wherein the step of compiling the shareable part of the first cursor to form a second cursor at the second node comprises:

generating a parse tree for an operation associated with said second cursor.

83.   (New) A computer-readable medium carrying one or more sequences of instructions for processing a request, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

initiating execution of a database statement at said first database server;

transmitting said database statement to a second database server prior to

termination of execution of said database statement at said first database server; and

generating, at said second database server, a structure required to prepare said database statement for execution at said second database server.

84.   (New) The computer-readable medium of claim 83, further comprising the step of receiving a request from a client at said second database server to execute said database statement.

85.   (New) The computer-readable medium of claim 84, further comprising the step of executing said database statement at said second database server using said structure.

86.   (New) The computer-readable medium of claim 83, wherein said structure is a parse tree.

87.   (New) The computer-readable medium of claim 83, wherein said structure is an execution plan.

21

88.   (New) The computer-readable medium of claim 83, wherein said step of transmitting said database statement to a second database server is performed in anticipation of a shutdown of said first database server.

89.   (New) The computer-readable medium of claim 83, wherein said step of transmitting said database statement to a second database server is performed prior to any indication that said first database server may become inoperable.

90.   (New) The computer-readable medium of claim 83, wherein said step of transmitting said database statement to a second database server comprises the step of:

determining that said second database server is capable of executing said database statement.

91.   A computer-readable medium carrying one or more sequences of instructions for processing a request, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

receiving a database statement to be executed at a first database server;

generating a structure required to prepare the statement for execution at said first database server;

initiating execution of said database statement at said first database server;

transmitting said structure to a second database server;

receiving a request to execute said database statement at said second database server; and

executing said database statement at said second database server using said structure.

92.   (New) The computer-readable medium of claim 91, wherein said structure is a parse tree.

93. (New) The computer-readable medium of claim 91, wherein said structure is an execution plan.

94. (New) The computer-readable medium of claim 91, wherein said step of transmitting said structure to a second database server is performed in anticipation of a shutdown of said first database server.

95. (New) The computer-readable medium of claim 91, wherein said step of transmitting said structure to a second database server is performed prior to any indication that said first database server may become inoperable.

96. (New) The computer-readable medium of claim 91, wherein said step of transmitting said structure to a second database server comprises the step of: determining that said second database server is capable of executing said database statement.

Docket No. 50277-1960 (OID 2000-195-01)